

**FIG. 1**  
PRIOR ART

```

1 c:\collections
2     notes.txt
3     myletter.doc
4     c-myhomepage
5
6     s
7         homepage.html
8         myphoto.jpg

```

**FIG. 2**

```

1 c:\collections
2     notes.txt
3     myletter.doc
4     c-myhomepage
5         cspec
6         s
7             homepage.html
8             myphoto.jpg

```

**FIG. 3**

```

1 collection      c-myhomepage
2 coll-type       cf-web-page
3 coll-desc       A sample homepage collection
4 end-collection

```

FIG. 4

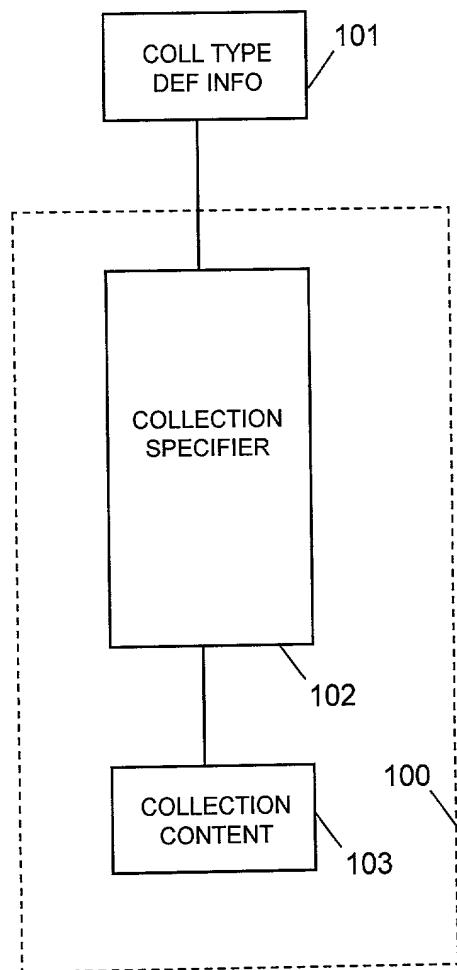


FIG. 5

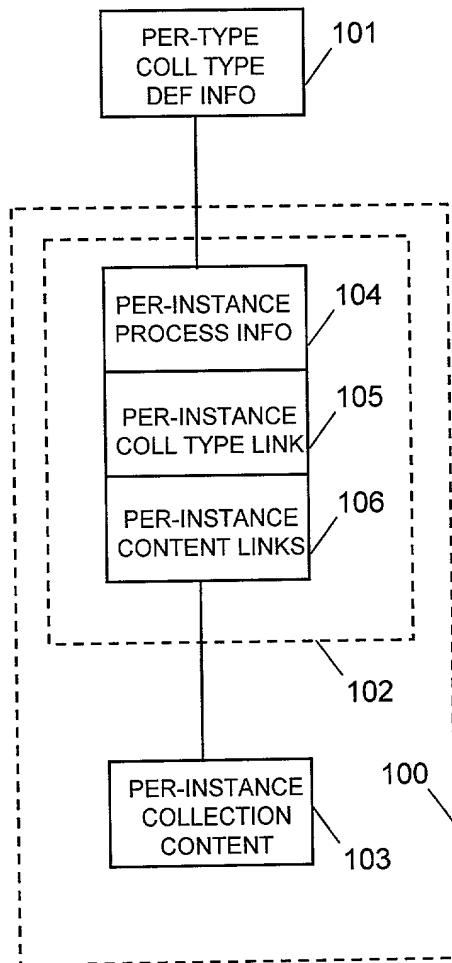


FIG. 6

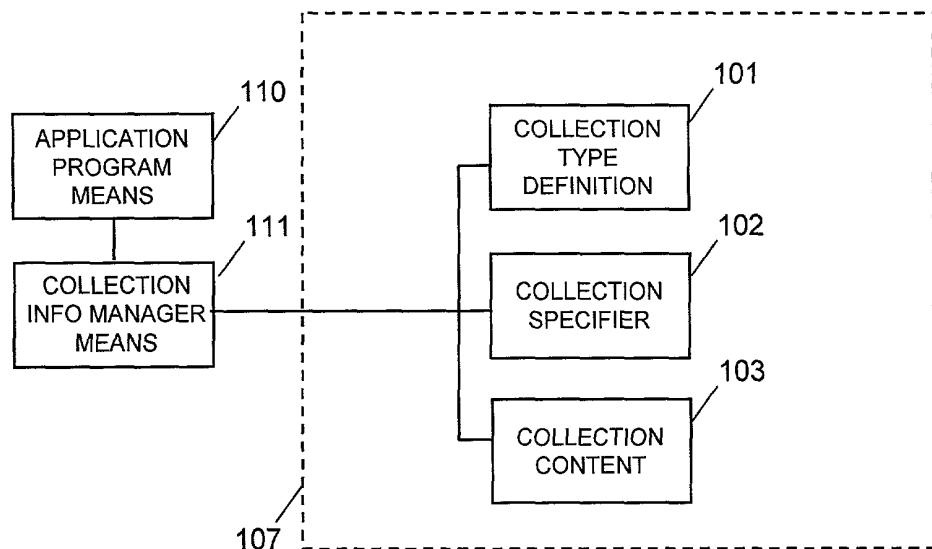
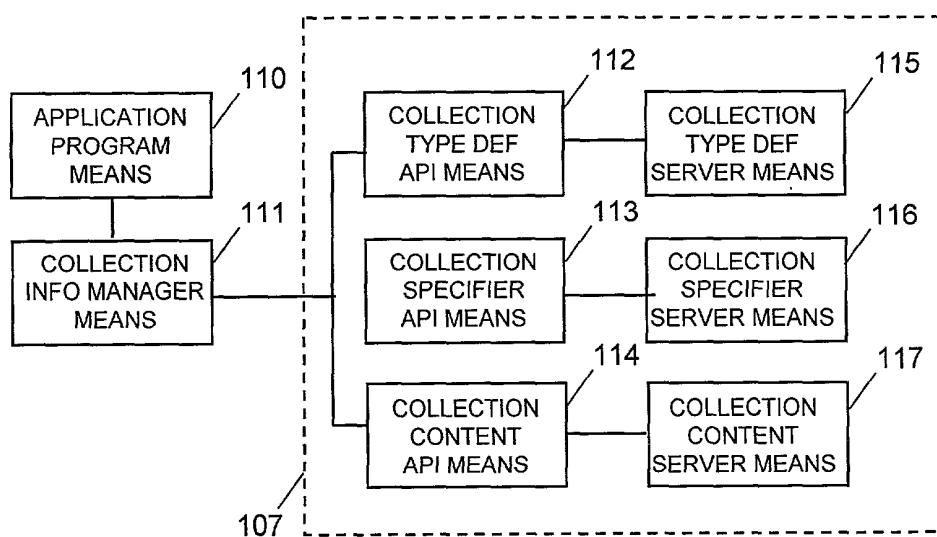


FIG. 7



4/25

## FIG. 8

```
1 /* collection data structure */
2 collection-info {

3     + specifier_info
4         + coll-type-indicator
5         + other specifier information ...
6
7     + content_info
8         + content_location_info ...
9         + content_members ...
10        + other content information...
11 }
```

## FIG. 9

```
1 /* collection type definition data structure */
2 collection-type-definition-info {

3     + coll-type-name
4     + collection internal structure info ...
5     + collection content location info ...
6     + collection content type recognition info ...
7
8 }
```

FIG. 10

<u>KEY</u>	<u>VALUE</u>
1 /* collection type internal structure definitions */	
2 dir_source_files	./s
3 dir_doc_files	./doc
4 /* content location definitions (per-type content links) */	
5 content_subtree_http	http://host.com/some/dir/name
6 content_subtree_ftp	ftp://host.com/some/dir/name
7 content_subtree_nfs	/some/local/directory/name
8 /* content type recognition definitions */	
9 content_policy	subtree_below_cspec_file
10 content_file_type	.c file_cpp
11 content_file_type	.c file_c
12 content_file_type	.h file_c_include
13 content_file_type	.doc file_ms_word
14 content_file_type	.html file_html
15 content_file_type	.xls file_ms_excel
16 /* collection processing definitions */	
17 compile_c_files	yes
18 compiler_windows	vc++
19 compiler_unix	gcc
20 build platforms	Win98, Win2000, gnulinux
21 process files	compile link
22 link libraries	stdio math sock
23 /* results dispatching definitions */	
24 results_ftp_host	ftp.output.com
25 results_ftp_dir	c:\ftphome\collection\results

6/25

FIG. 11

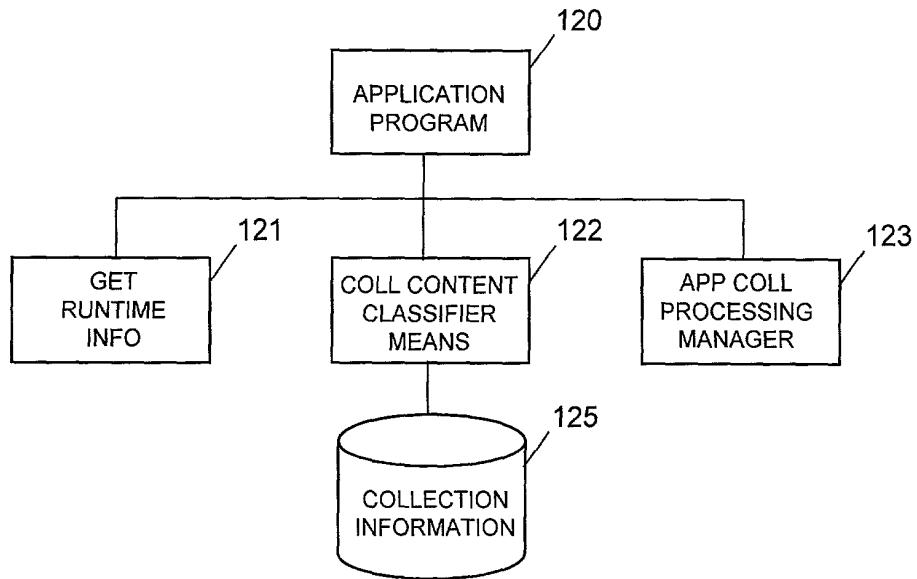
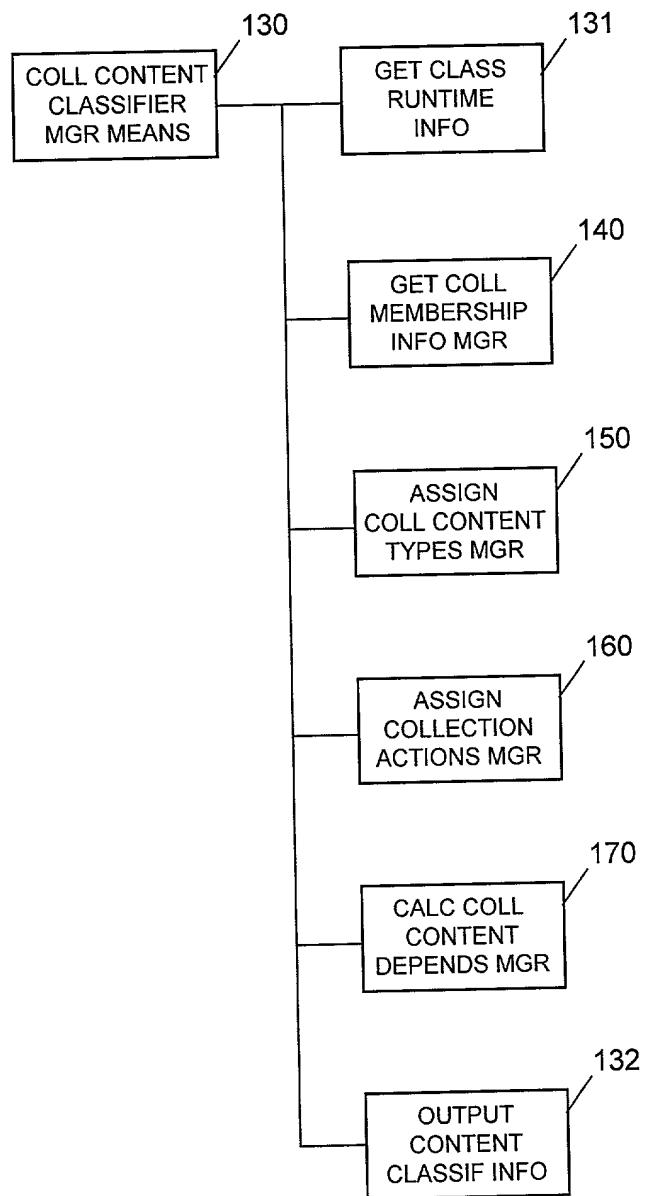


FIG. 12

- 1 /\* simplified algorithm for classifier-enabled program \*/
- 2 Call get runtime info to get collection to classify
- 3 Call collection content classifier to classify collection content
- 4 Call application processing functions to process the collection, using the content classification information produced in step 3.

7/25

FIG. 13



+

8/25

## FIG. 14

- 1 /\* simplified algorithm for classifier manager module \*/
- 2 Build data structures
- 3 Obtain runtime info including a collection to classify
- 4 Obtain lists of collection products and product contents
- 5 Assign a content type to each content member
- 6 Assign a symbolic action to each content member
- 7 Calculate dependencies for each content member
- 8 Organize and output information
- 9 Return classifier data structures to caller

## FIG. 15

```
1 /* top-level classified-content data structure */
2 coll-class-content {

3 /* classification info for this collection instance */
4     + collection type indicator
5     ... other collection info

6 /* classification info for this collection's products */
7     + list of prod-class-content structures
8     + list of content-class-entry structures

9 /* type definition info to support classification ops */
10    + collection type definition information
11    + product type definition information
12    + file type identification information
13    + content type definition information
14    + action type definition information

15 }
```

File Edit View Insert Tools Window Help

+

+

9/25

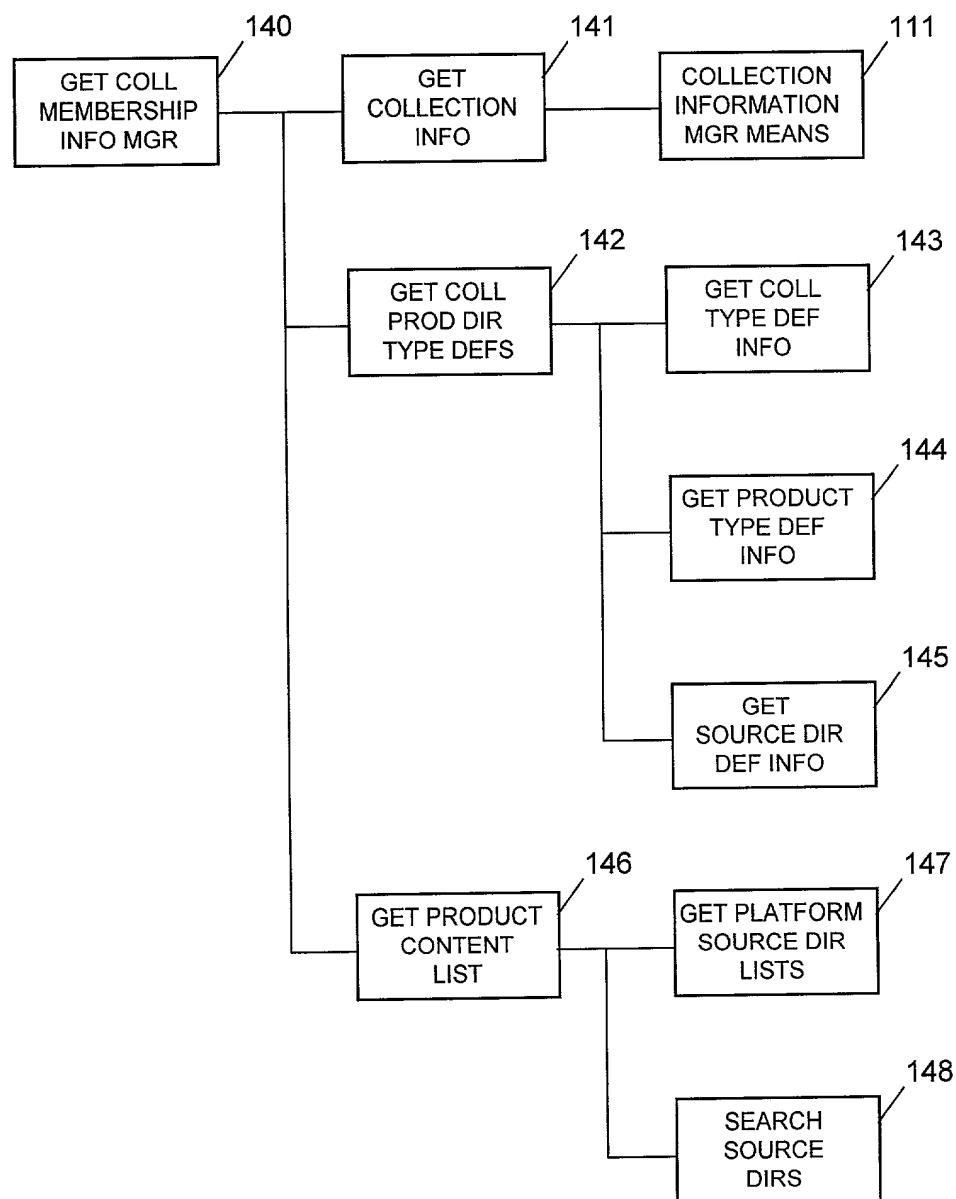
## FIG. 16

1	collection type definition information	
2	product type definition information	
3	content type definition information	
4	action type definition information	
5	cspec:	
6	coll-type	ct-program
7	index-coll-types.tbl:	
8	ct-program	ct-program.def
9	ct-web-page	ct-web-page.def
10	ct-program.def:	
11	product-type-index	index-product-types.tbl
12	index-product-types.tbl:	
13	pt-program	pt-program.def
14	pt-program.def:	
15	content-type-index	index-content-types.tbl
16	index-content-types.tbl	
17	content-c	content-c.def
18	content-c.def:	
19	action-type-index	index-action-types.tbl
20	index-action-types.tbl:	
21	action-c	action-c.def
22	action-c.def:	
23	parser-type	internal
24	parser-name	internal-c

+

10/25

FIG. 17





11/25

**FIG. 18**

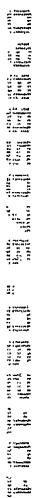
```
1 /* simplified algorithm for get coll membership info manager */
2 Build data structures

3 /* Obtain collection specifier info including coll product list */
4 Call Get Collection Specifier Info
5 Call Collection Information Manager

6 /* Obtain type definition info for coll type, product type, etc */
7 Call Get Collection Product Dir Type Information
8 Call Get Collection Type Definition Information
9 Call Get Product Type Definition Information
10 Call Get Product Source Directory Definition Information

11 /* Build per-product content lists from per-product src dir lists */
12 For each product:
13 Call Get Platform Source Directory List
14 Call Search Source Dirs to find content files in the source dirs
15 Add found content files to content-entry data structures

16 Return completed prod-class-content list data struct to caller
```



+

12/25

FIG. 19

```
1 /* prod-class-content data structure for one product */
2 prod-class-content {
3     + product type indicator
4     + list of product actions
5     + list of content-class-entry data structures
6     ... other product information
7 }
```

TOP SECRET//NOFORN

FIG. 20

```
1 /* content-class-entry data struct for one content entry */
2 content-class-entry {
3     + content entry pathname
4     + content type indicator
5     + content language type indicator
6     + content action indicator
7     + list of content dependencies
8     + ...
9 }
```

+

FIG. 21

```

1 index-coll-types.tbl:
2 ct-program           ct-program.def
3 ct-library           ct-library.def
4 ct-doc-html          ct-html.def

5 ct-program.def:
6 /* type definition info for a "ct-program" collection type */
7 product-type-index   index-prod-program.tbl
8 action-type-index    index-coll-action-types.tbl
9 action                cleanup
10 action               checkin
11 ... other collection type info

```

FIG. 22

```

1 index-prod-program.tbl:
2 pt-program           pt-program.def
3 pt-program-java      pt-program-java.def
4 pt-program-unix     pt-program-unix.def
5 pt-program-win       pt-program-win.def

6 pt-program.def:
7 /* type definition info for a "program" product type */
8 dir-source-files     source-dirs.lst
9 dir-library-files    library-dirs.lst
10 file-identification-table
11 content-type-index   index-content-types.tbl
12 action-type-index    index-prod-action-types.tbl
13 action                link
14 ... other product type info

```

14/25

### FIG. 23

```
1 source-dirs.lst:  
  
2 dir/gnulinux2      ./s/gnulinux2  
3 dir/gnulinux2      ./s/gnulinux  
4 dir/gnulinux2      ./s/unix  
5 dir/gnulinux2      ./s/pi  
6 dir/gnulinux2      ./s  
  
7 dir/win98          ./s/win98  
8 dir/win98          ./s/win  
9 dir/win98          ./s/pi  
10 dir/win98         ./s  
  
11 dir/win95         ./s/win95  
12 dir/win95         ./s/win  
13 dir/win95         ./s/pi  
14 dir/win95         ./s
```

### FIG. 24

```
1 library-dirs.lst:  
  
2 dir/gnulinux2      ./lib/gnulinux2  
3 dir/gnulinux2      ./lib/gnulinux  
4 dir/gnulinux2      ./lib/unix  
5 dir/gnulinux2      ./lib/pi  
6 dir/gnulinux2      ./lib  
  
7 dir/win98          ./lib/win98  
8 dir/win98          ./lib/win  
9 dir/win98          ./lib/pi  
10 dir/win98         ./lib
```

FIG. 25

```
1   c:\collections
2     c-mystuff
3       cspec

4       s
5         pi
6           file1.h
7           file1.c
8           win
9             file2-pd.h
10            file2-pd.c
11            win98
12              file2-pd.c
13            unix
14              file2-pd.h
15              file2-pd.c
16            gnulinux2
17              file2-pd.c

18       lib
19         pi
20           libfun.s.c
21           libfun.s2.c
22         win
23           libfun.s.h
24           libfun.s2.c
25         unix
26           libfun.s.h

27       win98.plt
28         ... compiled files, programs, libs, etc

29       gnulinux2.plt
30         ... compiled files, programs, libs, etc
```

16/25

FIG. 26

```
1  /* file list for win98 platform */
2  ./s/win98/file2-pd.c
3  ./s/win/file2-pd.h
4  ./s/pi/file1.h
5  ./s/pi/file1.c
6  ./lib/win/libfun.h
7  ./lib/win/libfun2.c
8  ./lib/pi/libfun.c

9  /* file list for gnulinux2 platform */
10 ./s/gnulinux2/file2-pd.c
11 ./s/unix/file2-pd.h
12 ./s/pi/file1.h
13 ./s/pi/file1.c
14 ./lib/unix/libfun.h
15 ./lib/unix/libfun2.c
16 ./lib/pi/libfun.c
```

FIG. 27

```
1 collection      c-my-example
2 coll-type       ct-program
3 coll-desc       A memdir and memfile example.
4 end-collection

5 product         myprog
6 prod-type       pt-program
7 memdir          c:\third-party\source-files
8 memdir-prepend  c:\search>this\dir\first
9 memdir-append   c:\search>this\dir\last
10 memfile         c:\third-party\product\somefile.obj
11 memfile-prepend c:\somewhere\file.c
12 memfile-append  c:\somewhere\file2.c
13 end-product
```

FIG. 28

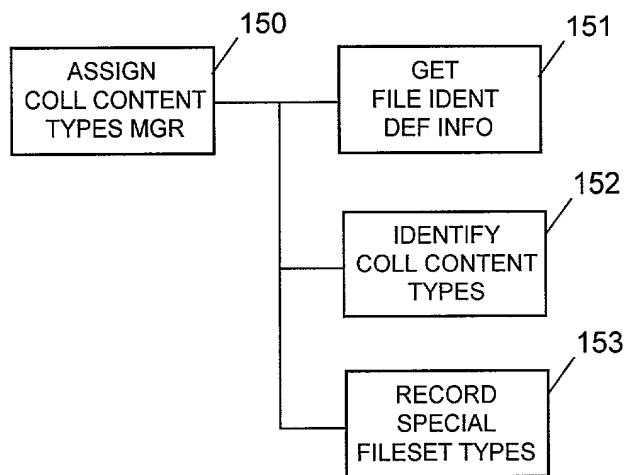


FIG. 29

- 1 /\* simplified algorithm for assign content type manager \*/
- 2 Build data structures
- 3 /\* Obtain file identification criteria to enable matching \*/
- 4 Call Get File Identification Definition Information
- 5 /\* Identify the content types of all files in the content list \*/
- 6 For each product in collection product list:
- 7     For each file in current product content list:
- 8         Call Identify Content Type to assign a content type
- 9 /\* Record special fileset types \*/
- 10 Call Record Special Fileset Types
- 11 Return data structures to caller

FIG. 30

```

1 file-identification.tbl:
2 /* Match-type    Match-string    Content-type */

3 exact          junkfile        ignore
4 suffix         .bak            ignore
5 tail           ~              ignore
6 content        #!/bin/csh      csh
7 suffix         .c              c-source
8 suffix         .h              c-header
9 suffix         .C             c++
10 suffix        .c++           c++

```

FIG. 31

```

1 collection      c-my-example
2 coll-type       ct-program
3 coll-desc       A fileset example.
4 end-collection

5 product         myprog
6 prod-type       pt-program

7 fileset         myset ..$/file1.f ..$/file2.f
8 fileset         myset ..$/file3.f
9 fileset-type    myset content-no-optimize

10 end-product

```

19/25

FIG. 32

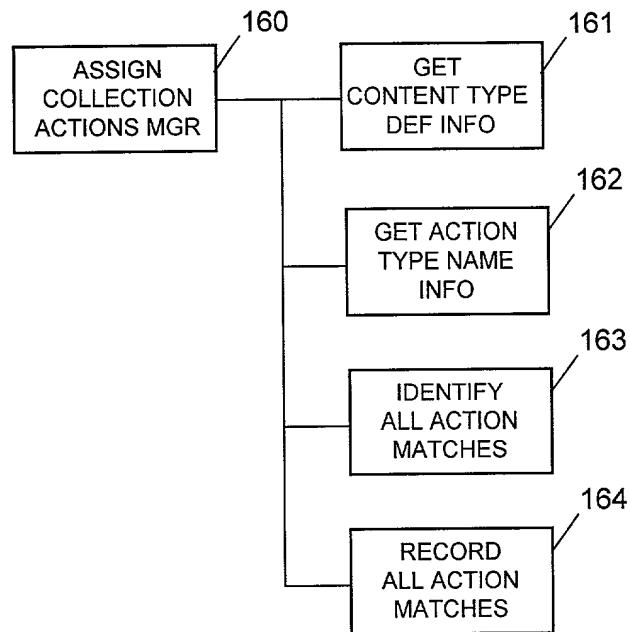


FIG. 33

- 1 /\* simplified algorithm for assign content actions manager \*/
- 2 Build data structures
- 3 /\* Obtain file type definition and action name index info \*/
- 4 Get File Type Definition Information
- 5 Get Action Type Name Information
- 6 /\* Assign action names to collection, product, and content entries \*/
- 7 Call Identify All Action Matches to choose coll, prod, content actions
- 8 Call Record All Action Matches to record choices in data structures
- 9 Return completed data structures to caller

FIG. 34

```

1 index-content-types.tbl:
2   c-source           content-c.def
3   c-header          content-c-h.def
4   csh              content-csh.def
5   html             content-html.def

6 content-c.def:
7 /* type definition info for a "c" file type */
8   type            c-source
9   language        c
10  action          action-c-source
11  action-type-index index-action-types.tbl
12 ... other content type definition info

```

FIG. 35

```

1 index-action-types.tbl:
2   action-c-source    action-c-source.def
3   action-c-header    action-c-header.def
4   action-csh         action-csh.def
5   action-html        action-html.def

6 action-c-source.def:
7   parser-type       internal
8   parser-name       internal-c

```

FIG. 36

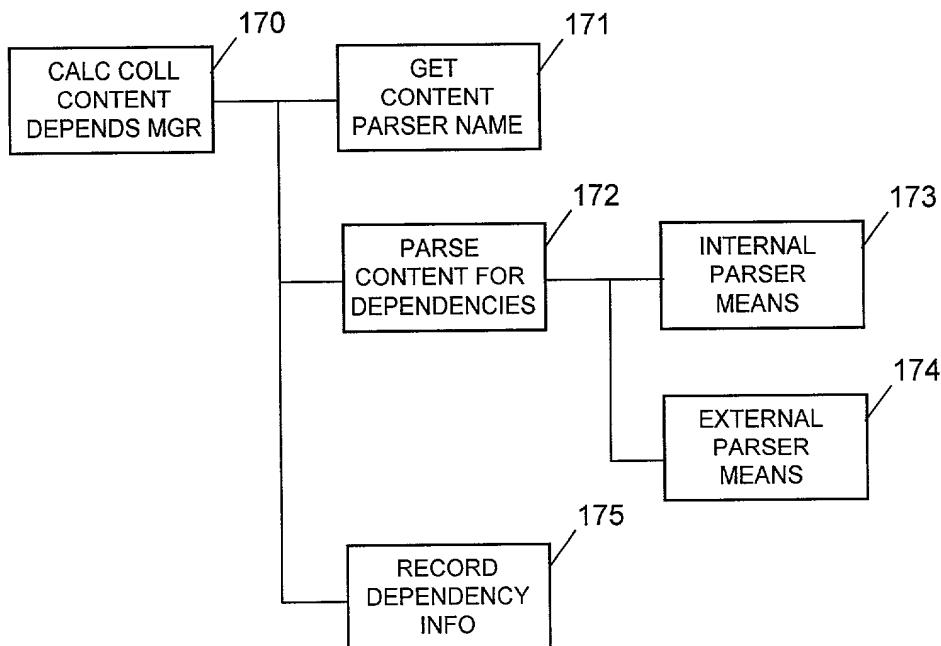


FIG. 37

- 1 /\* simplified algorithm for calc content dependencies mgr \*/
- 2 Build data structures
- 3 /\* loop over and parse all content entries \*/
- 4 For each product in collection product list:
- 5     For each file in current product content list:
- 6         Call Get Content Parser to choose a parser
- 7         Call Parse For Dependencies to obtain dependencies
- 8         Call Record Dependency Info to store dependencies in content-class-entry data structure
- 9 Return completed content-class-entry data structures to caller

FIG. 38

```
1 /* external parser invocation command line */
2 new-language-parser infile outfile
3 new-language-parser file1.c file1.cdeps

4 file1.cdeps:
5 /* dependency info from new-language-parser */
6 dep file1.c      stdio.h
7 dep file1.c      app-data.h
8 dep file1.c      content-class-entry.h
```

FIG. 39

```

1 c:\collections
2   c-my-example
3     cspec
4       s
5         pi
6           cmdline.h
7           win98
8           cmdline.c
9           gnulinux2
10          cmdline.c
11        lib
12          pi
13            libfuns.h
14            libfuns.c

```

FIG. 40

```

1 cspec:
2   collection  c-my-example
3   coll-type    ct-program
4   coll-desc   A multi-platform C program with library.
5   end-collection
6   product     myprog
7   prod-type   pt-program
8   prod-desc   A program product.
9   end-product
10  product    mylibrary
11  prod-type  pt-library
12  prod-desc  A library product.
13  end-product

```



FIG. 41

```
1  /* classification output for win98 platform */
2  collection          c-my-example
3  coll-type           ct-program
4  coll-action         cleanup
5  coll-action         checkin
6  ... other coll class info

7  /* classification info for a program product */
8  product             myprog
9  prod-type           pt-program
10 prod-action         link

11 content             cmdline.h
12 content-path        ..../s/pi/cmdline.h
13 content-type        c-header
14 content-language    c
15 content-action      none
16 end-content

17 content             cmdline.c
18 content-path        ..../s/win/cmdline.c
19 content-type        c-source
20 content-language    c
21 content-action      action-c-source
22 content-dep         ..../s/pi/cmdline.h
23 content-dep         ..../lib/pi/libfun.h
24 end-content

25 end-product
```

FIG. 42

```
1  /* classification output */
2  collection      c-my-example

3  ... /* classification info for the host collection */
4  ... /* classification info for the program product */

5  /* classification info for a library product */
6  product        mylibrary
7  prod-type     pt-library
8  prod-action   archive
9  prod-action   export-lib-for-sharing

10 content       libfun.h
11 content-path ..lib/pi/libfun.h
12 content-type c-header
13 content-language c
14 content-action export-incl-for-sharing
15 end-content

16 content       libfun.c
17 content-path ..lib/pi/cmdline.c
18 content-type c-source
19 content-language c
20 content-action action-c-source
21 content-dep   ..lib/pi/libfun.h
22 end-content

23 end-product
```